



An asymmetric 2D Pointer / 3D Ray for 3D Interaction within Collaborative Virtual Environments

Thierry Duval, Cédric Fleury

► To cite this version:

Thierry Duval, Cédric Fleury. An asymmetric 2D Pointer / 3D Ray for 3D Interaction within Collaborative Virtual Environments. Web3D 2009, Jun 2009, Darmstadt, Germany. pp.33-41, 10.1145/1559764.1559769 . inria-00433861

HAL Id: inria-00433861

<https://inria.hal.science/inria-00433861>

Submitted on 20 May 2010

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

An asymmetric 2D Pointer / 3D Ray for 3D Interaction within Collaborative Virtual Environments

Thierry Duval*

IRISA – Université de Rennes 1 – UEB – France

Cedric Fleury†

IRISA – INSA de Rennes – UEB – France

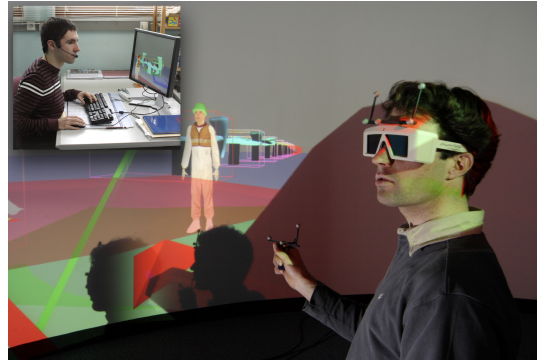


Figure 1: *Asymmetric use of the 2D Pointer / 3D Ray within a Collaborative Virtual Environment*

Abstract

In this paper we present a new metaphor for interaction within Collaborative Virtual Environments (CVE). This metaphor is dedicated to non-immersive or semi-immersive 3D interactions, for which users cannot afford to buy expensive devices neither for 3D visualization of their virtual environment nor for interaction. With these low-cost restrictions, we think that it is more effective to use basic 2D metaphors rather than to try to adapt 3D virtual metaphors which would be more difficult to use because of the poor immersion level offered by such systems.

The problem that will arise within a CVE is that it is difficult to make a user aware of the 2D metaphors used by another user, because they are not associated with a 3D virtual object of the shared universe. So our idea is to provide to a user a 3D virtual ray (using ray-casting for object selection) that would act like a 2D pointer on the screen, allowing the user to only control the 2D position of the closest ray end, and calculating the orientation of the ray so that its projection on the screen would always be a point. This way, since the user is controlling a 3D virtual ray, the other users can be made aware of his activity.

To test the efficiency of this 2D Pointer / 3D Ray, we have made some experiments making users compare different devices to realize some simple selection and manipulation tasks. The results show that this kind of 2D solution is efficient and allows 3D interaction within Virtual Environments by people who cannot afford expensive immersive hardware. This new metaphor allows more users to collaborate within CVE.

*e-mail: thierry.duval@irisa.fr

†e-mail: cedric.fleury@irisa.fr

CR Categories: H.5.1 [Information Interfaces and Presentation (e.g. HCI)]: Multimedia Information Systems—Artificial, augmented, and virtual realities; I.3.6 [Computer Graphics]: Methodology and Techniques—Interaction techniques

Keywords: 3D Interaction, 2D Pointer, 3D Ray, Ray-Casting, Collaborative Virtual Environments

1 Introduction

What is the best technical solution for easy and natural 3D interaction within Virtual Environments (VE)? Most people will answer that it is immersion, but to obtain high quality immersion you need stereovision for the visualization, linked to a 3D tracking device in order to track the position of tools of the user and of her head. Indeed, such technical solutions allow the images to be generated such that virtual tools can be colocated with parts of the user's body or with the real tools she is using, so a user feels like her arms, hands, or tools were really embedded within the virtual environments. Furthermore, interaction metaphors that are usually used in this context, such as virtual hands [Poupyrev et al. 1996], virtual rays [Bowman and Hodges 1997] or virtual 3D cursors [Zhai et al. 1994], are interesting for Collaborative Virtual Environments (CVE) because they provide a natural 3D representation that is perceptible for the other users of the CVE. Due to this 3D visualization of the interaction tools, a user can be easily aware of the activity of the other users of the CVE. Nevertheless, 2D metaphors and input devices have also to be considered for 3D interactions because they are sometimes easier to use than 3D metaphors, as stated in [Bowman et al. 2008].

However, a good immersion cannot be obtained without expensive hardware such as high-frequency video-projectors (for active stereovision) or double projectors (for passive polarized stereovision). Providing only stereovision is not enough to obtain a good immersion, because it cannot ensure a good colocation between the virtual tools driven by the users and the physical objects or body parts that the user uses to control the virtual tools. We need wireless tracking systems (optical, ultrasonic or magnetic) for head tracking, tools tracking and body parts tracking.

Without colocation, we consider that it would be difficult for somebody to use efficiently the classical 3D interaction metaphors, and that these metaphors will not be user-friendly. So perhaps basic 2D interaction tools such as a 2D pointer driven with a classical 2D mouse could be as efficient as the usual 3D metaphors for simple tasks such as object selection and manipulation (3D positioning, for instance).

Two problems arise when using such basic 2D interaction metaphors. First, when several users share a CVE, it will be difficult to make a user aware of the interactions of other users, because their 2D interaction tools will not be associated with any 3D virtual objects. Second, using a classical mouse will not fit semi-immersive environments when a user stands in front of a big image produced by a videoprojector, generally without any keyboard or 2D mouse.

This is the reason why we propose a new 2D pointer that will be associated with a 3D geometry in order to appear visually within the Virtual Environment. This 2D pointer will be easy to use and will be driven by any device that can control a 2D position: for example a classical 2D mouse, a gamepad or a Nintendo wiimote remote gaming controller. The 3D geometry of this pointer will be a virtual ray, so other users can be easily made aware of the movement of this 3D ray, in the same way they can be made aware of the evolution of classical 3D interaction metaphors. This 2D Pointer / 3D Ray will use the classical ray-casting technique for object selection and manipulation. In this way, its behavior is similar to the aperture based selection technique [Forsberg et al. 1996] and to the technique developed in [Ware and Lowther 1997].

In order to show that our 2D Pointer / 3D Ray can be useful for selection and basic interaction tasks, we have made some experiments comparing four interaction techniques. We will describe the conditions of the experiments, then we will present the results of these experiments and we will discuss them in order to show that our new interaction metaphor is efficient enough to be used for interaction within CVE when some of the users do not have access to expensive immersive hardware devices.

2 Related Work

2.1 3D interaction metaphors

The “3D Interaction Techniques for 3D Manipulation” chapter of [Bowman et al. 2004] presents many metaphors dedicated to 3D manipulation. The ray-casting [Poupyrev et al. 1998] is very interesting because it is very simple and efficient, especially when used at close range. This metaphor is difficult to use at long range because it requires high angular accuracy, but some approaches allow to minimize jitter, such as adaptive control display gains (e.g. pointer acceleration) [Frees et al. 2007]. Driving such a metaphor through a 2D input device allows the reduction of jitter around the orientation of the virtual ray, especially when the 2D device is a mouse [Balakrishnan et al. 1997]. The remaining problem is that rotating a 3D ray using a 2D input device is not user-friendly. This is why we propose to adapt this metaphor so that it can be easily driven with a device that only provides a 2D position.

2.2 Awareness within CVE

It is important for people sharing a CVE to be aware of the activity of other users, as explained in [Fraser, M. et al. 1999], in order to help them to understand the evolution of the CVE and to collaborate more efficiently with the other users. Showing the activity of a user to the other users with whom he may collaborate is a central point for an efficient collaboration, a lot of work has been realized

in this area [Fraser, M. et al. 2000][Gutwin, C. and Greenberg, S. 1998]. Many egocentric metaphors, such as the virtual ray casting, are well suited for interaction within CVE, thanks to their graphical visualization that can be shown to the other users.

3 The asymmetric 2D Pointer / 3D Ray

Our idea is to use a 3D virtual ray that would be as easier to drive than the classical 2D mouse pointer. The result looks like a classical 2D pointer moving on the surface of the screen. In fact it is a quite thin and long 3D virtual ray, moving near the viewpoint of the user, staying always at the same depth, which orientation is calculated in a way that its projection on the screen is always a small spot.

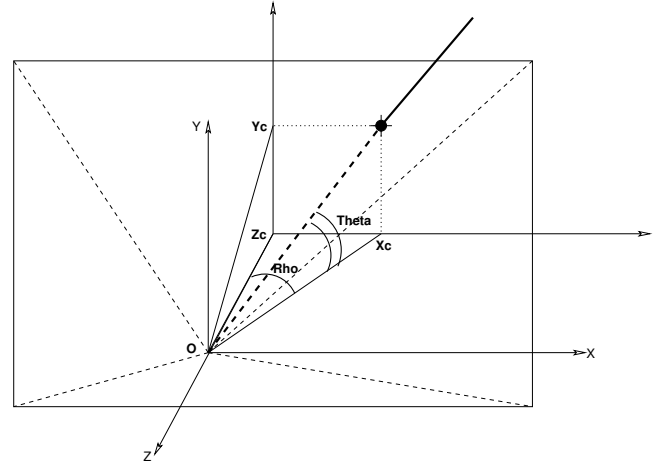


Figure 2: Projection of the 3D Ray as a small spot on the screen.

As shown on figure 2, the 2D device used to control the pointer will provide the X_c and the Y_c values, and the Z_c value is a chosen one, so the ρ and θ values can be calculated this way, if the ρ angle (the heading) is first applied around the Y axis and then the θ angle (the elevation) is applied around the X' axis :

- $\rho = \text{atan}(-X_c/Z_c)$
- $\theta = \text{atan}(Y_c/\sqrt{X_c * X_c + Z_c * Z_c})$

This way, the user of the 2D Pointer / 3D Ray will always feel that she is using a 2D pointer (figure 3), while other users will see a 3D virtual ray moving thanks to the action of the first user (figure 4). So it is quite easy to use by the first user, and quite easy to understand by the other users.

This 2D Pointer / 3D Ray is completely independent from the hardware device that will be used to drive it: either a classical 2D mouse, or a game pad, or any device able to provide 2D coordinates, or even a graphical 2D user interface.

As the 2D Pointer / 3D Ray is turning around the closest extremity of the virtual ray, the movements of a manipulated object can also be affected by a small rotation and will not stay at the same Z coordinate within the user's coordinate system, except if we force it to preserve its relative orientation and Z coordinate.

This metaphor can be easily extended to 3D movements within the user's coordinate system: the X and the Y coordinates are directly provided by the physical device used to drive the 2D pointer, and the Z coordinate can be changed by moving the manipulated object along the 3D ray. To achieve such a translation along the virtual ray, the device used to drive the 2D pointer must also provide the

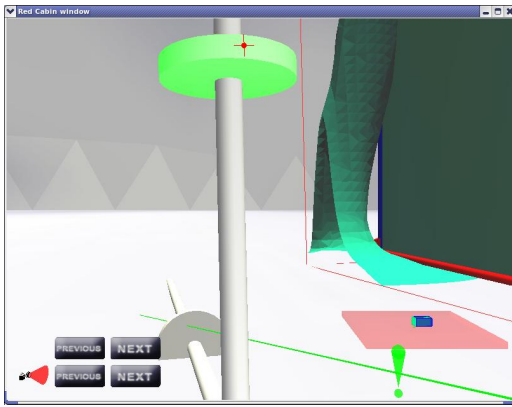


Figure 3: User 1 moves a 3D slider with her red 2D pointer and she sees the green 3D virtual ray of user 2 ready to select another slider.

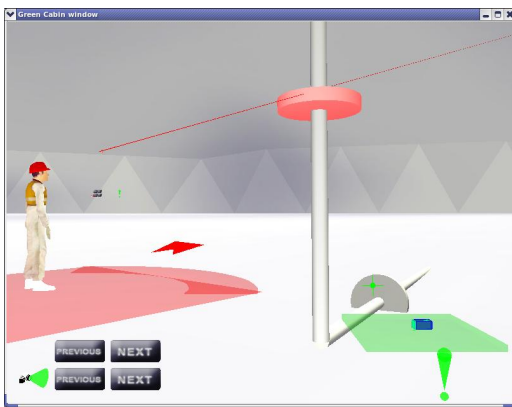


Figure 4: User 2 is ready to select a slider with her green 2D pointer while she is looking at user 1 moving a slider with her red 3D virtual ray.

information needed to calculate the Z coordinate, or it can be associated to another device providing this value. For example, this Z coordinate can be obtained thanks to the wheel of a 2D mouse, or some buttons of a gamepad.

A rotation of the manipulated object within the user's coordinate system can also be calculated with additional devices, for example the keyboard or some buttons or joysticks of a gamepad.

We consider our technique as an egocentric interaction metaphor using a pointer as described in [Poupyrev et al. 1998]. As our 2D Pointer / 3D Ray is a tool associated to the user's viewpoint, the user carries this interaction tool with her when she navigates within the VE, in the same manner as 3DM [Butterworth et al. 1992]. So as the 2D Pointer / 3D Ray moves with the viewpoint when the user navigates, the object that has been grabbed by the moving tool navigates also within the VE, which is another complementary way to provide a new position and orientation to this manipulated object.

Last, the 2D Pointer / 3D Ray can simply be used as a classical 2D pointer to trigger some elements of a 3D GUI that could be carried by the user, in order to control the state of the application.

So, according to Hand [Hand 1997] who separates virtual interactions into 3 categories: 3D interaction (selection and manipulation), navigation and application control; we see that our 2D Pointer / 3D

Ray, carried by the user, is well suited for these three kinds of interactions.

4 Hypotheses

We think that our 2D Pointer / 3D Ray can be quite efficient for 3D manipulation, at least for simple tasks such as positioning objects within a VE, especially when we cannot offer a good immersion to the user. So we will make some hypotheses about the usability of our interaction metaphor, then we will make some experiments to verify these hypotheses.

4.1 H1: the best solution for 3D interaction is immersion with head-tracking

This solution will be used as a reference for our evaluation. We will compare the time spent during the manipulation and the accuracy of the other evaluated techniques relative to this one. We will also ask the users which technical solution they prefer: we think they will prefer immersion with head-tracking and colocation of the virtual ray with the interaction device they use.

4.2 H2: our 2D Pointer / 3D Ray can be as efficient than immersion with head-tracking

We hope that this hypothesis will be true, at least for basic tasks such as positioning 3D objects, to be able to propose it to a user when we cannot offer him any immersion. This would also offer to new VR users one efficient interaction tool which is as easy to use as the classical mouse and its associated 2D pointer.

4.3 H3: immersion without head-tracking is not a good solution for 3D interaction

We think that incomplete immersion is not a good solution, because users can feel deceived by interactive solutions that would work much better if virtual tools could be colocated with real tools or with body parts of the user. This solution should not be neither as fast nor as accurate as a solution offering total immersion. This solution could even be the worst because it denatures the 3D metaphors it uses. Indeed, in this case the user is nearly placed in an exocentric situation, as she cannot really use the egocentric metaphors in the way they should be used, as it is not possible here to collocate a virtual ray and its associated interaction device.

4.4 H4: in a semi-immersive context, our 2D Pointer / 3D Ray can be as efficient as 3D interaction without head-tracking

This semi-immersive situation appears quite often, and can easily be obtained thanks to a simple videoprojector, which is quite affordable today. In this context, the user is generally standing in front of a big projection screen, so she cannot use neither the mouse nor the keyboard for 3D interaction within the VE. So our idea is to provide to this user our 2D Pointer / 3D Ray, driven with another kind of device: a wireless one in order to allow the user to move easily. Here we propose to use a Nintendo wiimote to drive our interaction metaphor. We hope that this low-cost solution (which costs around 50 US dollars) can be nearly as efficient as an optical 3D tracking technology (which costs roughly fifty thousand dollars).

5 The experiments

In order to verify our hypotheses, we have set up a simple experiment that consists in positioning four 3D objects (colored spheres) within their four associated supports (semi-transparent colored cylinders). We ask the users to realize the task as fast as possible, with the best accuracy possible.

5.1 The tasks to complete

We make the user aware of the selection of a 3D object by overlaying a semi-transparent upscaled geometry, and we also give them information about the accuracy of the manipulation task: a flag, associated to each support, changes its color from red towards green when the position of its associated object is accurate enough to consider that the positioning task is completed.

5.1.1 All the objects are at the same depth relative to the user

For the first manipulations, in order to make it easy for the user, all the 3D objects (the spheres and their associated cylindric supports) are at the same depth relative to the user. The positioning task is only a 2D task, and we do not allow the user to change the position of his viewpoint by navigating.

The user must complete this task three times: first with big objects, second with medium objects, and third with small objects. The experiment set-up is shown in figures 5, 6 and 7.

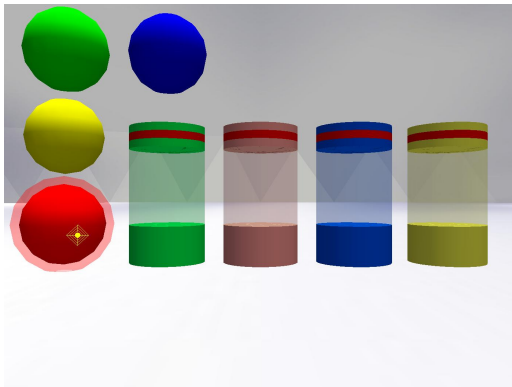


Figure 5: Big objects at the same distance from the user.

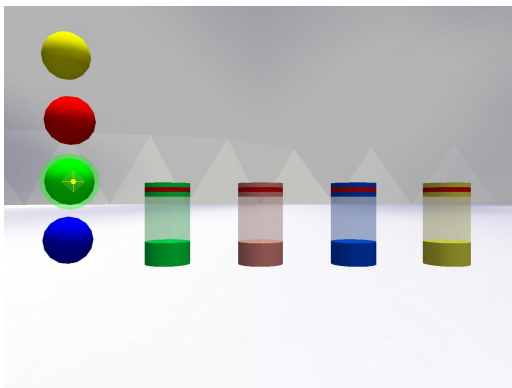


Figure 6: Medium objects at the same distance from the user.

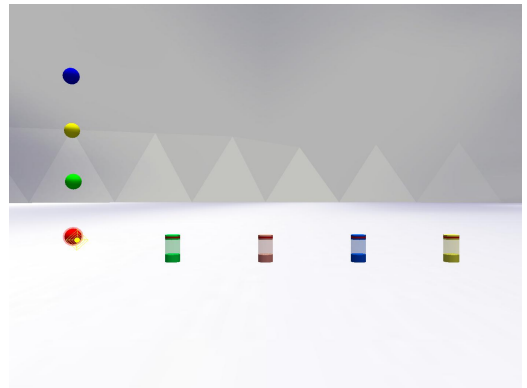


Figure 7: Small objects at the same distance from the user.

5.1.2 All the objects are at different depths relative to the user

Second, we place the 3D objects at different depths relative to the user: there is not any 3D object to manipulate that is located at the same depth as its associated support. So the user must grab the 3D objects and move them (also along the front/back axis) to place each object into its support.

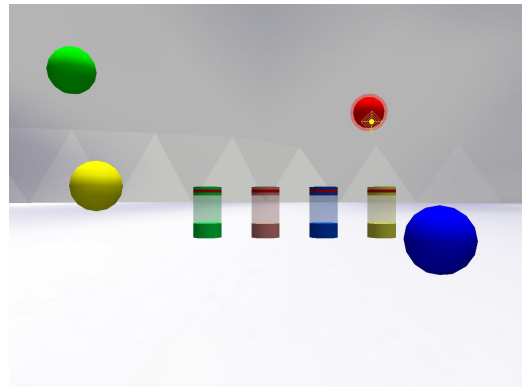


Figure 8: Medium objects at different distances from the user.

Here again, the user must complete this task three times: first with big objects, second with medium objects, and third with small objects. The experiment set-up for medium objects is shown on figure 8.

If the navigation is not accurate enough, as we use the same navigation step for the three sizes configurations, users will have to adjust the depth in a way proposed by the current interaction technique they are testing.

5.2 The four interaction techniques to compare

Each user has to complete these six positioning tasks with four hardware configurations, from non-immersive context with a simple mouse in front of a workstation, to fully immersive context with stereovision, 3D tool tracking and head-tracking.

For these four configurations, we will use the same device for the front/back navigation: the joystick of the Nintendo nunchuk extension of the Nintendo wiimote, and only the front/back information will be used to constrain the user to stay on this front / back axis.

So the user will not be allowed to navigate in order to come near every object: he will have to be quite far from the objects placed on the right and on the left, otherwise there would not have been significant differences due to the objects sizes between these three experimental setups.

5.2.1 Technique 1: the 2D mouse used as a 2D pointer in front of a simple screen

First, the classical 2D mouse is used to drive the 2D Pointer / 3D Ray and the user sits in front of a 20" screen. The 2D Pointer / 3D Ray is perceived here as a simple 2D pointer associated with the usual mouse pointer. We use the left button press of the mouse to grab an object located "under" the pointer, and the release of this button releases the grabbed object. An object which is grabbed by the pointer can have its Z coordinate (its depth relative to the user viewpoint) adjusted by using the wheel of the mouse.

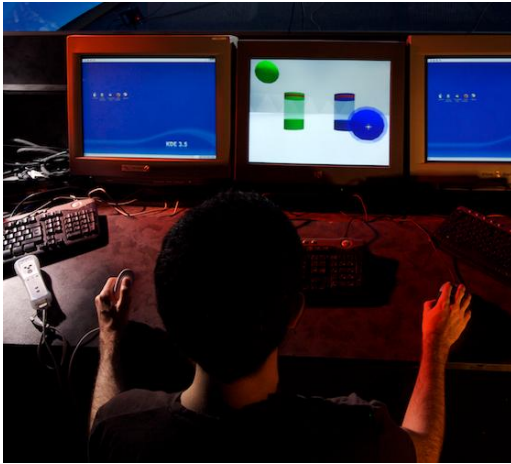


Figure 9: Technique 1: the 2D mouse used as a 2D Pointer.

5.2.2 Technique 2: the Nintendo wiimote used as a 2D pointer in front of a big videoprojection

Second, a Nintendo wiimote is used to drive the 2D Pointer / 3D Ray and the user stands in front of a 2.5 meters high - 3 meters wide projection. The 2D coordinates are acquired thanks to the infrared camera of the Nintendo wiimote, pointing at an infrared emitter placed between the user and the projection. Here again, the 2D Pointer / 3D Ray is perceived as a simple 2D pointer. We use the press of the "A" or "B" button of the Nintendo wiimote to grab an object located "under" the pointer, and the release of one of these buttons releases the grabbed object. An object which is grabbed by the pointer can have its Z coordinate (its depth relative to the user's viewpoint) adjusted with successive presses of the "Up" and "Down" buttons of the Nintendo wiimote.

5.2.3 Technique 3: optical tracking for the 3D Ray in front of a wide videoprojection with stereovision

Third, we use an optical 3D tracking system to acquire the 3D position and orientation of a Nintendo wiimote that is used as a tool in the hand of the user. This 3D position, relative to the viewpoint, is used to control a 3D virtual ray. The user stands in front of a 2.5 meters high and 9 meters wide semi-cylindric projection screen, offering stereovision. As for the second technique, we still use the press of the "A" or "B" button of the Nintendo wiimote to grab an object traversed by the 3D virtual ray, and the release of one of these

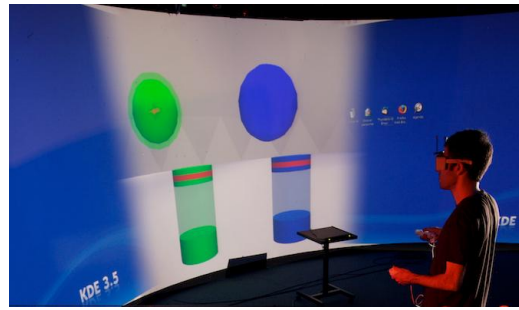


Figure 10: Technique 2: the Nintendo wiimote used as a 2D Pointer.

buttons releases the grabbed object. An object which is grabbed by the 3D virtual ray can have its 3 position coordinates (including its depth relative to the user's viewpoint) affected by the position and the orientation of the 3D virtual ray, which can be much more important than in the case of the manipulation of the 2D Pointer / 3D Ray. Here the user will adjust the Z relative position of a grabbed object by moving the Nintendo wiimote front or back. With this third technique, the infrared camera of the Nintendo wiimote is not used any more, but we still use the Nintendo wiimote for its "A" and "B" buttons.

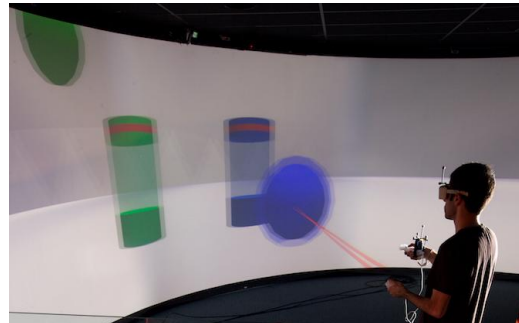


Figure 11: Technique 3: optical tracking for the 3D Ray.

5.2.4 Technique 4: optical tracking for the 3D Ray and the head of the user in front of a wide videoprojection with stereovision

Fourth and last, we extend Technique 3 to use the optical 3D tracking system not only to acquire the 3D position and orientation of a Nintendo wiimote that is used as a tool in the hand of the user, but also to acquire the 3D position and orientation of the head of the user, to be able to compute the best image for him. These 3D positions enable the position and orientation of a 3D virtual ray to be computed and to produce the most appropriate image in order to make the user believe that the virtual ray is placed at the exact end of her Nintendo wiimote.

5.3 Completing the tasks

Each user had to complete the 6 tasks in the same order :

1. big objects at the same depth,
2. medium objects at the same depth,
3. small objects at the same depth,

4. big objects at different depths,
5. medium objects at different depths,
6. small objects at different depths.

These 6 tasks had to be completed for each of the 4 techniques, always in the same order : Technique 1, Technique 2, Technique 3, Technique 4.

We did so because we were thinking that the users could then encounter the 4 techniques beginning with the simplest one and finishing with the most complicated one, so the first two techniques could serve as a practice for the last two techniques.

Most of the results show that we were right to make this assumption because the best manipulation times are those obtained with the first technique.

Before doing the real tasks, the users had a few minutes to practice each of the 4 techniques with a simple task: only 2 big objects with their supports, a first object with the same depth than the supports, and the second object with a different depth, in order to make the user try the navigation with the nunchuk.

5.4 Test user demograph

34 people completed our experiments. 94% were men. Their average age was 26.5 years old. Almost everybody had experience with computers and 2D interaction with the mouse. Around 50% of the users had already used 3D interaction. 20% of the users had already played 3D video games. Most of the users were computer science students, software engineers or computer science researchers or teachers.

6 Results

6.1 Raw results

For each task (placement of 4 objects), we measured the time needed to complete the task, the accuracy of the positioning of the objects, and the average of the selections needed to grab an object and to release it at the correct position.

6.1.1 Time spent

Figure 12 shows the average time spent (in seconds) to complete each task with each technique.

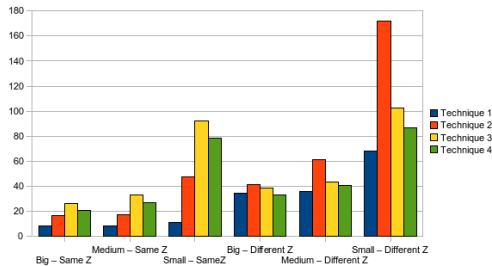


Figure 12: Time needed to complete the tasks.

A single factor ANOVA on the participants' time was performed for the six tasks, then for the 2D positioning tasks only, and last for the 3D positioning tasks only, each time with Technique 4 as a reference. The global ANOVA indicated that the difference in participants' time was significant for Technique 1 ($F(1, 814) = 20.52$,

$p < 0.0001$) and Technique 2 ($F(1, 814) = 6.81$, $p = 0.0092$), but not really for Technique 3 ($F(1, 814) = 3.53$, $p = 0.0609$). The second ANOVA (2D tasks) indicated that the difference in participants' time was significant for Technique 1 ($F(1, 404) = 65.93$, $p < 0.0001$), Technique 2 ($F(1, 404) = 13.61$, $p = 0.0003$), and for Technique 3 ($F(1, 404) = 4.58$, $p = 0.0328$). The third ANOVA (3D tasks) indicated that the difference in participants' time was significant only for Technique 2 ($F(1, 404) = 29.70$, $p < 0.0001$), and not for neither Technique 1 ($F(1, 404) = 1.16$, $p = 0.28$) nor Technique 3 ($F(1, 404) = 1.32$, $p = 0.25$). Nevertheless, a Student test performed between Technique 3 and Technique 4 indicated that their difference was significant enough ($t = 2.2782$, $p = 0.02325$).

The preliminary conclusions are that:

1. For 2D positioning tasks:
 - Technique 1 is the fastest technique (H2 partially verified for Technique 1).
 - Technique 3 is the worst one (H3 partially verified).
 - Technique 2 is better than Techniques 3 and 4 (H4 partially verified, H2 partially verified for Technique 2).
2. For 3D positioning tasks:
 - Technique 1 is better than Technique 4 (H2 partially verified for Technique 1 and H1 partially verified), but not very significantly.
 - Technique 4 is the second most efficient one, but the difference with the best Technique is not very significant (H1 partially verified).
 - Technique 3 is slower than technique 4 (H3 partially verified), but not very significantly.

6.1.2 Accuracy

Figure 13 shows the average relative error for each task with each technique.

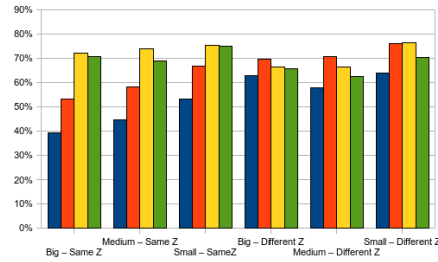


Figure 13: Error percentages.

A single factor ANOVA on the participants' precision was also performed for the six tasks with Technique 4 as a reference. It indicated that the difference in participants' precision was significant for Technique 1 ($F(1, 814) = 109.83$, $p < 0.0001$), Technique 2 ($F(1, 814) = 4.84$, $p = 0.0283$), and Technique 3 ($F(1, 814) = 4.20$, $p = 0.0407$). A Student test performed between Technique 3 and Technique 4 indicated also that their difference was significant ($t = 2.22$, $p = 0.02697$).

The preliminary conclusions about accuracy are nearly the same than about speed :

1. Technique 1 is always the most accurate technique (H0 verified, H2 verified for Technique 1).
2. For 3D positioning tasks, Technique 4 is second (H1 partially verified).
3. Technique 3 is always less accurate than technique 4 (H1 and H3 partially verified).
4. For 2D positioning tasks, Technique 3 is the worst (H3 partially verified).
5. For 2D positioning tasks, Technique 2 is better than Techniques 3 and 4 (H4 partially verified, H2 partially verified also for Technique 2).
6. For 3D positioning tasks, Technique 2 is quite as good as Technique 3 (H4 partially verified).

6.1.3 Average number of selections per object

Figure 14 shows the average number of selection per object needed to complete each task with each technique.

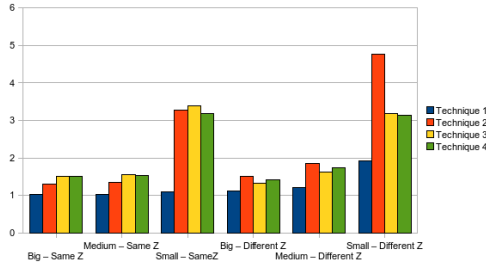


Figure 14: Average selections per object to complete a task.

A single factor ANOVA on the participants' number of selections was performed for the six tasks, then for the 2D positioning tasks, and last for the 3D positioning tasks, each time with Technique 4 as a reference.

The global ANOVA indicated that the difference in participants' precision was significant for Technique 1 ($F(1, 814) = 482.24, p < 0.0001$) and Technique 2 ($F(1, 814) = 7.89, p = 0.0051$), but not for Technique 3 ($F(1, 814) = 0.40, p = 0.5231$), which was confirmed by a Student test performed between Technique 3 and Technique 4 ($t = 0.6793, p = 0.4973$).

The preliminary conclusions are only that:

1. Technique 1 is always the most direct technique (H2 verified for Technique 1).
2. Techniques 2, 3 and 4 are equivalent except for small objects with different depths where Technique 2 is not good.

6.2 Subjective results

Last, we asked each user to give a relative ordering, then an absolute scoring, of this 4 techniques according to :

1. How pleasant was it ?
2. How easy was it to use ?
3. How fast was it to use ?
4. How accurate was it to use ?
5. How efficient was it to use ?

6. How tiring was it to use ?

For the relative ordering, users had to give 1 to the best technique and 4 to the worst.

For the absolute scoring, users had to give 1 if the technique was very good, 2 if it was good, 3 if it was acceptable, and 4 if it was not good.

6.2.1 Relative ordering of the 4 techniques

Figure 15 shows how the users ordered the 4 techniques we wanted to compare.

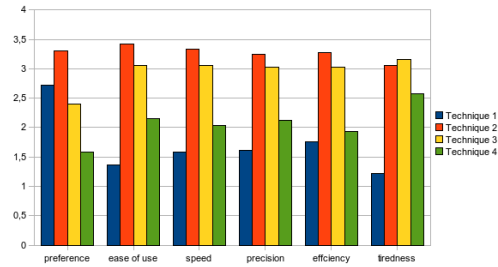


Figure 15: Relative ordering of the techniques.

The preliminary conclusions are that:

1. Technique 1 is the easiest, the fastest, the most accurate, the most efficient and the least tiring of the fourth techniques (H2 verified for Technique 1).
2. Technique 4 is the preferred one and the second one concerning performances (H1 partially verified).
3. Technique 3 is always worse than technique 4 (H1 and H3 partially verified).
4. Techniques 2 and 3 have very similar results.

We notice that people have quite a bad opinion of Technique 2, even for speed and accuracy, although the time and accuracy effectively measured for this technique were not so bad, except for the last task to complete, with small objects at different depths. No doubt that this task influenced the votes. We will discuss this point later.

6.2.2 Absolute scoring of the 4 techniques

Figure 16 shows how the users scored the 4 techniques we wanted to evaluate.

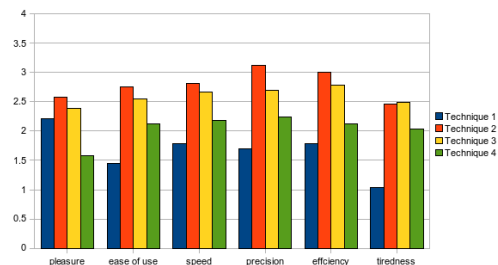


Figure 16: Absolute scoring of the techniques.

The preliminary conclusions are the same as those about the relative ordering of the 4 techniques, which is consistent.

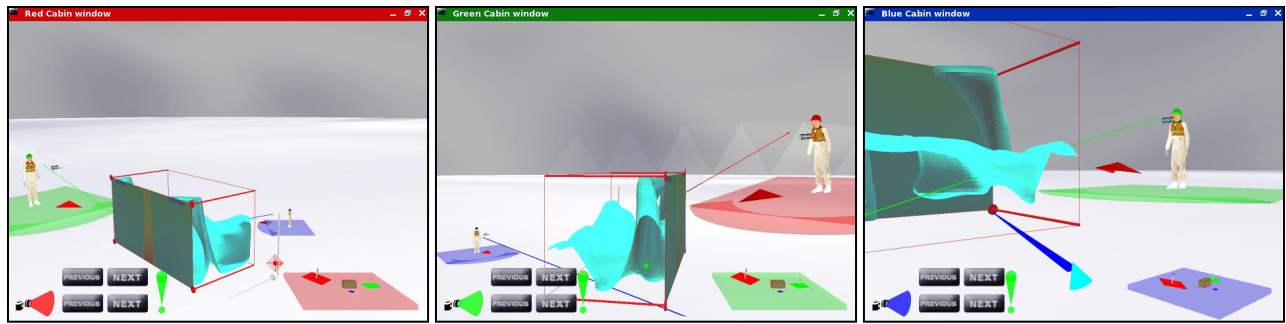


Figure 17: 3 users interacting within a CVE: left and center users use a 2D Pointer / 3D Ray while right user uses a 3D virtual ray

7 Discussion

Now let us examine our hypotheses to see if they were verified.

7.1 H1: the best solution for 3D interaction is immersion with head-tracking

Yes. Technique 4 was the second best for speed and accuracy, according to the measures, and the preferred technique according to the votes of the users. Most of the users really enjoyed this technique, and succeeded in using it quite efficiently even if it was the first time they had used stereovision and head-tracking. Nevertheless, some users signaled that this technique can be quickly tiring.

7.2 H2: our 2D Pointer / 3D Ray can be as efficient than immersion with head-tracking

Yes. It is even more efficient, especially when it is driven with a simple 2D mouse in front of a workstation. The Technique 1 is the fastest and the more accurate one, according to the measures, and also according to the votes of the users. It will allow a user, without immersion capabilities, to interact in a CVE with our 2D Pointer / 3D Ray that will be perceived as a 3D virtual ray by the other users.

7.3 H3: immersion without head-tracking is not a good solution for 3D interaction

Nearly yes. Technique 3 is always worse than Techniques 1 and 4, and many users told us that Technique 3 was lacking “something”, which was “head-tracking” once they had experienced Technique 4. Technique 2 is better than Technique 3 for 2D positioning, and nearly as good for 3D positioning except for small objects with different depths.

7.4 H4: in a semi-immersive context, our 2D Pointer / 3D Ray can be as efficient as 3D interaction without head-tracking

Nearly yes. Considering the raw results, Technique 2 is the second best for 2D manipulations, but the slowest technique as soon as the user must make depth adjustments. Considering the subjective results, Technique 2 is the technique users do not like, but it does not make such a big difference with technique 3 for the absolute scoring. We think that these results can be explained because we did not choose the best way to make the depth adjustments for 2D positioning: it was quite difficult for the users to press the “Up” and “Down” buttons whilst simultaneously keeping the “A” or “B” button. The Nintendo wiimote suffered additional difficulties because

of the optical targets that were attached for optical tracking, making it difficult to reach the wiimote buttons (the same Nintendo wiimote with the optical targets was used for the experiments of Techniques 2, 3 and 4), as shown figure 18. Last, Z movements were discrete in this setup whereas the Z movements were continuous in the three other setups.



Figure 18: Optically tracked Nintendo wiimote

8 Conclusion

We have proposed a new metaphor for 3D interaction within Collaborative Virtual Environments: the 2D Pointer / 3D Ray, which associates a 3D representation with a 2D pointing device (for example a 2D mouse). This metaphor allows an asymmetric collaboration between users immersed within a CVE (via stereovision and head-tracking) and users simply sitting in front of the screen of their workstation. The user without immersion will interact as easily as if he had a simple 2D pointer, as the associated 3D ray (a 3D virtual ray) will be continuously moved and oriented in a way that its projection on the screen of the user will always be a small spot. The other users of the CVE will be made aware of the action of this user thanks to the movements of his associated 3D virtual.

We have made some experiments to confirm the usability of our interaction metaphor, comparing it to usual 3D interaction with a virtual ray with stereovision and head-tracking for colocation. As we were assuming that our metaphor is as well adapted to collaboration as the classical 3D ray-casting technique, we have only made experiments about its stand-alone usability. The results show that this metaphor is very easy to use with a 2D mouse for 3D positioning tasks. It is more difficult to drive the metaphor with a Nintendo wiimote especially for depth adjustments of small objects. However, we feel that this kind of device can be very helpful to provide interaction tools in semi-immersive environments as described in

our experiments. This device has been used to provide interaction facilities in the context of 3D exploration of scientific data [Duval et al. 2008] as illustrated figure 17. A first solution to improve the depth adjustment is to adapt the size of the tool to the size of the objects we have to manipulate, by changing the size of the user relatively to the size of the world as proposed in [Mine et al. 1997]. We could even determine the ideal size automatically [Kopper et al. 2006] by placing the correct information within the object we want to manipulate.

9 Future work

There are several other ways to improve the depth adjustment with a device such as the Nintendo wiimote. The immediate solution is to use its infrared camera, which is able to provide the position of several targets. This allows to calculate the distance between the 2 targets of our sensor-bar acquired by the camera, and use the variation of this distance to propose a depth adjustment (as it is used in the snooker game of the "Wiplay" pack). This solution is already operational but has yet to be experimented. Another immediate solution is to change the gain of the joystick of the nunchuk used for navigation, it would work in the same way as changing the size of the interaction tool relatively to the size of the world. Other solutions can also be proposed by using the accelerometers of the wiimote.

We think that this kind of device will be easier to use if it can also control the orientation of a selected object, although full 3D rotations are not always necessary as explained in [Bowman et al. 2008]. So, we should realize some experiments in order to test if our metaphor can be extended to 3D orientation of 3D objects. A first solution is to associate the 2D mouse and keyboard modifiers to switch from a "depth adjustment" mode towards several "axis rotation" modes. Another solution is to take a similar approach with gamepads or with the Nintendo wiimote, combining buttons and 2D pointing or depth adjustment to provide rotation around a chosen axis.

Acknowledgements

This work has been initiated in a joint of the ANR N° 06 TLOG 031 Part@ge project and of the ANR N° 06 TLOG 029 SCOS project. It is now going on in the ANR-08-COSI-003-01 CollaViz project.

References

- BALAKRISHNAN, R., BAUDEL, T., KURTENBACH, G., AND FITZMAURICE, G. 1997. The Rockin'Mouse: Integral 3D Manipulation on a Plane. In *CHI '97: Proceedings of the SIGCHI conference on Human factors in computing systems*, ACM, New York, NY, USA, 311–318.
- BOWMAN, D. A., AND HODGES, L. F. 1997. An evaluation of techniques for grabbing and manipulating remote objects in immersive virtual environments. In *SI3D '97: Proceedings of the 1997 symposium on Interactive 3D graphics*, ACM Press, New York, NY, USA, 35–ff.
- BOWMAN, D. A., KRUIJFF, E., LAVIOLA, J. J., AND POUPYREV, I. 2004. *3D User Interfaces: Theory and Practice*. Addison Wesley Longman Publishing Co., Inc., Redwood City, CA, USA.
- BOWMAN, D., COQUILLART, S., FROELICH, B., HIROSE, M., KITAMURA, Y., KIYOKAWA, K., AND STUERZLINGER, W. 2008. 3D User Interfaces: New Directions and Perspectives. *Computer Graphics and Applications, IEEE 28*, 6, 20–36.
- BUTTERWORTH, J., DAVIDSON, A., HENCH, S., AND OLANO, M. T. 1992. 3DM: a Three Dimensional Modeler using a Head-Mounted Display. In *SI3D '92: Proceedings of the 1992 symposium on Interactive 3D graphics*, ACM Press, New York, NY, USA, 135–138.
- DUVAL, T., FLEURY, C., NOUAILHAS, B., AND AGUERRECHE, L. 2008. Collaborative Exploration of 3D Scientific Data. In *VRST '08: Proceedings of the 2008 ACM symposium on Virtual Reality Software and Technology*, ACM, New York, NY, USA, 303–304.
- FORSBERG, A., HERNDON, K., AND ZELEZNIK, R. 1996. Aperture based selection for immersive virtual environments. In *UIST '96: Proceedings of the 9th annual ACM symposium on User interface software and technology*, ACM, New York, NY, USA, 95–96.
- FRASER, M., BENFORD, S., HINDMARCH, J., AND HEATH, C. 1999. Supporting Awareness and Interaction through Collaborative Virtual Interfaces. *UIST'99, Asheville, USA*, 27–36.
- FRASER, M., GLOVER, T., VAGHI, I., BENFORD, S., GREENHALGH, C., HINDMARCH, J., AND HEATH, C. 2000. Revealing the Realities of Collaborative Virtual Reality. *CVE'2000, San Francisco*, 29–37.
- FREES, S., KESSLER, G. D., AND KAY, E. 2007. PRISM interaction for enhancing control in immersive virtual environments. *ACM Trans. Comput.-Hum. Interact.* 14, 1, 2.
- GUTWIN, C., AND GREENBERG, S. 1998. Design for Individuals, Design for Groups: Tradeoffs Between Power and Workspace Awareness. *CSCW'98, Seattle, Washington, US*, 207–216.
- HAND, C. 1997. A Survey of 3D Interaction Techniques. *Computer Graphics Forum* 16, 5, 269–281.
- KOPPER, R., NI, T., BOWMAN, D. A., AND PINHO, M. 2006. Design and Evaluation of Navigation Techniques for Multiscale Virtual Environments. In *VR'06: Proceedings of the IEEE Virtual reality Conference*, IEEE Computer Society, Washington, DC, USA, 175–182.
- MINE, M. R., FREDERICK P. BROOKS, J., AND SEQUIN, C. H. 1997. Moving Objects in Space: Exploiting Proprioception in Virtual-Environment Interaction. In *SIGGRAPH '97: Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, ACM Press/Addison-Wesley Publishing Co., New York, NY, USA, 19–26.
- POUPYREV, I., BILLINGHURST, M., WEGHORST, S., AND ICHIKAWA, T. 1996. The Go-Go Interaction Technique: Non-Linear Mapping for Direct Manipulation in VR. In *UIST '96: Proceedings of the 9th annual ACM symposium on User interface software and technology*, ACM Press, New York, NY, USA, 79–80.
- POUPYREV, I., WEGHORST, S., BILLINGHURST, M., AND ICHIKAWA, T. 1998. Egocentric Object Manipulation in Virtual Environments: Empirical Evaluation of Interaction Techniques. *Computer Graphics Forum* 17, 3.
- WARE, C., AND LOWTHER, K. 1997. Selection Using a One-eyed Cursor in a Fish Tank VR Environment. *ACM Trans. Comput.-Hum. Interact.* 4, 4, 309–322.
- ZHAI, S., BUXTON, W., AND MILGRAM, P. 1994. The "Silk Cursor": Investigating Transparency for 3D Target Acquisition. In *CHI '94: Proceedings of the SIGCHI conference on Human factors in computing systems*, ACM Press, New York, NY, USA, 459–464.